Necessary Conditions for Tractability of Valued CSPs*

Johan Thapper
Université Paris-Est, Marne-la-Vallée, France
thapper@u-pem.fr

Stanislav Živný
University of Oxford, UK
standa.zivny@cs.ox.ac.uk

Abstract

The connection between constraint languages and clone theory has been a fruitful line of research on the complexity of constraint satisfaction problems. In a recent result, Cohen et al. [SICOMP'13] have characterised a Galois connection between valued constraint languages and so-called weighted clones. In this paper, we study the structure of weighted clones. We extend the results of Creed and Živný from [CP'11/SICOMP'13] on types of weightings necessarily contained in every nontrivial weighted clone. This result has immediate computational complexity consequences as it provides necessary conditions for tractability of weighted clones and thus valued constraint languages. We demonstrate that some of the necessary conditions are also sufficient for tractability, while others are provably not.

1 Introduction

The constraint satisfaction problem (CSP) is a general framework capturing decision problems arising in many contexts of computer science [1, 16, 21]. The CSP is NP-hard in general but there has been much success in finding tractable fragments of the CSP by restricting the types of relation allowed in the constraints. A set of allowed relations has been called a *constraint language* [18, 25]. For some constraint languages the associated constraint satisfaction problems with constraints chosen from that language are solvable in polynomial-time, whilst for other constraint languages this class of problems is NP-hard [18, 26]; these are referred to as *tractable languages* and *NP-hard languages*, respectively. Dichotomy theorems, which classify each possible constraint language as either tractable or NP-hard, have been established for constraint languages over two-element domains [40], three-element domains [6], for conservative (containing all unary relations) constraint languages [8], for maximal constraint languages [5, 9], for graphs (corresponding to languages containing a single binary symmetric relation) [20], and for digraphs without sources and sinks (corresponding to languages containing a single binary relations without sources and sinks) [3]. The most successful approach to classifying the complexity of constraint languages has been the algebraic approach [2, 7, 26].

The *valued* constraint satisfaction problem (VCSP) is a generalisation of the CSP that captures not only decision problems but also optimisation problems [12, 24, 50]. A VCSP instance associates with each constraint a *weighted relation*, which is a $\overline{\mathbb{Q}}$-valued function, where $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ is the set of extended rational numbers, and the goal is to minimise the sum of the weighted relations associated with all constraints. Tractable fragments of the VCSP have been identified by restricting the types of allowed weighted relations that can be used to define the valued constraints. A set of allowed weighted relations has been called a *valued constraint language* [12]. Classifying the complexity of *all* valued constraint languages is a challenging task as it includes as a special case the

*Stanislav Živný was supported by a Royal Society University Research Fellowship.

classification of $\{0,\infty\}$-valued languages (i.e., constraint languages), which would answer the conjecture of Feder and Vardi [18], which asserts that every constraint language is either tractable or NP-hard, and its algebraic refinement, which specifies the precise boundary between tractable and NP-hard languages [7]. However, several nontrivial results are known. Dichotomy theorems, which classify each possible valued constraint language as either tractable or NP-hard, have been established for valued constraint languages over two-element domains [12], for conservative (containing all $\{0,1\}$-valued unary cost functions) valued constraint languages [34], and for minimum-solution languages (containing relations and a single unary injective weighted relation) [47]. Furthermore, it has been shown that a dichotomy for constraint languages implies a dichotomy for valued constraint languages [32]. Moreover, the power of the basic linear programming relaxation [31,33,45] and the power of the Sherali-Adams relaxations [47] for valued constraint languages have been characterised.

Cohen et al. have recently introduced an algebraic theory of weighted clones [10] for classifying the computational complexity of valued constraint languages. This theory establishes a one-to-one correspondence between valued constraint languages closed under expressibility (which does not change the complexity of the associated class of optimisation problems), called weighted relational clones, and weighted clones [10]. This is an extension of (part of) the algebraic approach to CSPs which relies on a one-to-one correspondence between constraint languages closed under pp-definability (which does not change the complexity of the associated class of decision problems), called relational clones, and clones [7], thus making it possible to use deep results from universal algebra.

Creed and Živný initiated the study of weighted clones and have used the theory of weighted clones to determine certain necessary conditions on nontrivial weighted clones and thus on tractable valued constraint languages [13], see also [10]. In particular, [13] simplifies the NP-hardness part of the complexity classification of Boolean valued constraint languages from [12].

Contributions We continue the study of weighted clones started in [10,13]. After introducing valued constraint satisfaction problems and all necessary tools in Section 2, we study, in Section 3, structural properties of nontrivial weighted clones. Our main result on weighted clones, Theorem 5, is an extension of a result from [10] that provides a more fine-grained characterisation of what conditions on weighted clones are necessary for tractability. Moreover, we demonstrate that some of the necessary conditions are also sufficient for tractability, while others are provably not. Overall, we give a structural result that shows what types of weightings are guaranteed to exist in nontrivial weighted clones. As a direct consequence, we narrow down the possible structure of tractable weighted clones. A proof of our main result is presented in Section 4 and is based on an application of Gordan's theorem, which is a variant of LP duality. The introduced technique is novel and might prove useful in future work on weighted clones. Finally, we relate our results to *maximal* tractable valued constraint languages, or equivalently, to *minimal* tractable weighted clones.

Related work Given the generality of the VCSP, there have been results on the complexity of special types of VCSPs. Finite-valued CSPs are VCSPs in which all weighted relations are $\mathbb{Q}$-valued. In other words, finite-valued CSPs are purely optimisation problems and thus do not include as a special case (decision) CSPs. The authors have recently classified all finite-valued constraint languages on arbitrary finite domains [46]. Minimum Solution (Min-Sol) problems are Valued CSPs with one unary *injective* $\mathbb{Q}$-valued weighted relation and $\{0,\infty\}$-valued weighted relations. Min-Sols generalise Min-Ones [14] and bounded integer linear programs. Min-Sols have been only very recently classified [47] with respect to computational complexity, thus improving on previous

partial classifications [27–30, 48]. Minimum Cost Homomorphism (Min-Cost-Hom) problems are Valued CSPs in which all but unary weighted relations are $\{0,\infty\}$-valued. Thus the optimisation part of the problem is only given by a sum of unary terms. This may seem very restrictive but it is known [11, 39] that any VCSP is equivalent to a VCSP where only the (not necessarily injective) unary constraints involve optimisation. Min-Cost-Hom problems with all unary cost functions have been classified in [43]. Also, Min-Cost-Hom problems with all unary $\{0,\infty\}$-valued cost functions [44, 48] and on three-element domains [49] have been classified.

2 Preliminaries

2.1 Valued CSPs

Throughout the paper, let D be a fixed finite set of size at least two.

Definition 1. *An m-ary* relation *over D is any mapping $\phi : D^m \to \{c,\infty\}$ for some $c \in \mathbb{Q}$. We denote by $\mathbf{R}_D^{(m)}$ the set of all m-ary relations and let $\mathbf{R}_D = \bigcup_{m\geq 1} \mathbf{R}_D^{(m)}$.*

An m-ary relation over D is commonly defined as a subset of D^m. Note that Definition 1 is equivalent to the standard definition as any subset of D^m can be associated with the set $\{\mathbf{x} \in D^m \mid \phi(\mathbf{x}) < \infty\}$. Consequently, we shall use both definitions interchangeably.

Given an m-tuple $\mathbf{x} \in D^m$, we denote its ith entry by $\mathbf{x}[i]$ for $1 \leq i \leq m$.

Let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers with (positive) infinity.

Definition 2. *An m-ary* weighted relation[1] *over D is any mapping $\phi : D^m \to \overline{\mathbb{Q}}$. We denote by $\mathbf{\Phi}_D^{(m)}$ the set of all m-ary weighted relations and let $\mathbf{\Phi}_D = \bigcup_{m\geq 1} \mathbf{\Phi}_D^{(m)}$.*

From Definition 2 we have that relations are a special type of weighted relations. If needed we call a weighted relation *unweighted* to emphasise the fact that ϕ is a relation.

Example 1. An important example of a (weighted) relation is the binary equality $\phi_=$ on D: $\phi_=(x,y) = 0$ if $x = y$ and $\phi_=(x,y) = \infty$ if $x \neq y$.

For any m-ary weighted relation $\phi \in \mathbf{\Phi}_D^{(m)}$, we denote by $\mathrm{Feas}(\phi) = \{\mathbf{x} \in D^m | \phi(\mathbf{x}) < \infty\} \in \mathbf{R}_D^{(m)}$ the underlying *feasibility relation*.

A weighted relation $\phi : D^m \to \overline{\mathbb{Q}}$ is called *finite-valued* if $\mathrm{Feas}(\phi) = D^m$.

Definition 3. *Let $V = \{x_1, \ldots, x_n\}$ be a set of variables. A* valued constraint *over V is an expression of the form $\phi(\mathbf{x})$ where $\phi \in \mathbf{\Phi}_D^{(m)}$ and $\mathbf{x} \in V^m$. The number m is called the* arity *of the constraint, the weighted relation ϕ is called the* constraint weighted relation*, and the tuple $\mathbf{x}$ the* scope *of the constraint.*

We call D the *domain*, the elements of D *labels* (for variables), and say that the weighted relation in $\mathbf{\Phi}_D$ take *values*.

Definition 4. *An instance of the* valued constraint satisfaction problem *(VCSP) is specified by a finite set $V = \{x_1, \ldots, x_n\}$ of variables, a finite set D of labels, and an* objective function *I expressed as follows:*

$$I(x_1, \ldots, x_n) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i)\,, \tag{1}$$

[1] In some paper weighted relations are called cost functions.

where each $\phi_i(\mathbf{x}_i)$, $1 \leq i \leq q$, is a valued constraint over V. Each constraint can appear multiple times in I.

The goal is to find an assignment *(or a* labelling*) of labels to the variables that minimises I.*

CSPs are a special case of VCSPs with (unweighted) relations with the goal to determine the existence of a feasible solution.

Example 2. The Max-Cut problem for a graph is to find a cut with the largest possible size. This problem is NP-hard [19] and equivalent to the Min-UnCut problem with respect to exact solvability. For a graph (V, E) with $V = \{x_1, \ldots, x_n\}$, this problem can be expressed as the VCSP instance $I(x_1, \ldots, x_n) = \sum_{(i,j)\in E} \phi_{\mathsf{xor}}(x_i, x_j)$ over the Boolean domain $D = \{0, 1\}$, where $\phi_{\mathsf{xor}} : \{0, 1\}^2 \to \overline{\mathbb{Q}}$ is defined by $\phi_{\mathsf{xor}}(x, y) = 1$ if $x = y$ and $\phi_{\mathsf{xor}}(x, y) = 0$ if $x \neq y$.

Definition 5. *Any set $\Gamma \subseteq \mathbf{\Phi}_D$ is called a* valued constraint language[2] *over D, or simply a* language*. We will denote by* VCSP(Γ) *the class of all VCSP instances in which the constraint weighted relations are all contained in Γ.*

Definition 6. *A valued constraint language Γ is called* tractable *if* VCSP(Γ') *can be solved (to optimality) in polynomial time for every finite subset $\Gamma' \subseteq \Gamma$, and Γ is called* intractable *if* VCSP(Γ') *is NP-hard for some finite $\Gamma' \subseteq \Gamma$.*

A valued constraint language is called *finite-valued* if every weighted relation ϕ from the language is finite-valued. Example 2 shows that the finite-valued constraint language $\{\phi_{\mathsf{xor}}\}$ is intractable.

We denote by $\mathrm{Feas}(\Gamma) = \{\mathrm{Feas}(\phi) | \phi \in \Gamma\}$ the set of underlying relations of all weighted relations from Γ.

2.2 Weighted relational clones

Definition 7. *We say that an m-ary weighted relation ϕ is* expressible *over a valued constraint language Γ if there exists a VCSP instance $I \in$ VCSP(Γ) with variables $V = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$, such that*

$$\phi(y_1, \ldots, y_m) = \min_{x_1\in D, \ldots, x_n\in D} I(x_1, \ldots, x_n, y_1, \ldots, y_m)\,. \tag{2}$$

A valued constraint language Γ is *closed under expressibility* if every weighted relation ϕ expressible over Γ belongs to Γ.

Definition 8. *A valued constraint language $\Gamma \subseteq \mathbf{\Phi}_D$ is called a* weighted relational clone *if it contains the binary equality relation $\phi_=$ on D and is closed under expressibility, scaling by non-negative rational constants (where we define $0 \cdot \infty = \infty$), and addition of rational constants.*

For any Γ, we define wRelClone(Γ) *to be the smallest weighted relational clone containing Γ.*

Note that for any weighted relational clone Γ if $\phi \in \Gamma$ then $\mathrm{Feas}(\phi) \in \Gamma$ as $\mathrm{Feas}(\phi) = 0\phi$.

Definition 9. *A* relational clone *is a weighted relational clone containing only (unweighted) relations.*[3] *For a set of relations Γ, we denote by* RelClone(Γ) *the smallest relational clone containing Γ.*

It has been shown that Γ is tractable if and only if wRelClone(Γ) is tractable [10]. Consequently, when trying to identify tractable valued constraint languages, it is sufficient to consider only weighted relational clones.

[2] A valued constraint language Γ is sometimes called *general-valued* to emphasise the fact that weighted relations from Γ are not necessarily finite-valued.

[3] Equivalently, a set of relations containing the binary equality relation and closed under conjunction and existential quantification.

2.3 Weighted clones

Any mapping $f : D^k \to D$ is called a k-ary *operation*. We will apply a k-ary operation f to k m-tuples $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^m$ coordinatewise, that is,

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = (f(\mathbf{x}_1[1], \ldots, \mathbf{x}_k[1]), \ldots, f(\mathbf{x}_1[m], \ldots, \mathbf{x}_k[m]))\,. \tag{3}$$

Definition 10. *Let ϕ be an m-ary weighted relation on D and let f be a k-ary operation on D. Then f is a* polymorphism *of ϕ if, for any $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in D^m$ with $\mathbf{x}_i \in \mathrm{Feas}(\phi)$ for all $1 \le i \le k$, we have that $f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) \in \mathrm{Feas}(\phi)$.*

For any valued constraint language Γ over a set D, we denote by $\mathrm{Pol}(\Gamma)$ the set of all operations on D which are polymorphisms of all $\phi \in \Gamma$. We write $\mathrm{Pol}(\phi)$ for $\mathrm{Pol}(\{\phi\})$.

A k-ary *projection* is an operation of the form $\mathrm{e}_i^{(k)}(x_1, \ldots, x_k) = x_i$ for some $1 \le i \le k$. Projections are (trivial) polymorphisms of all valued constraint languages.

Definition 11. *The* superposition *of a k-ary operation $f : D^k \to D$ with k ℓ-ary operations $g_i : D^\ell \to D$ for $1 \le i \le k$ is the ℓ-ary function $f[g_1, \ldots, g_k] : D^\ell \to D$ defined by*

$$f[g_1, \ldots, g_k](x_1, \ldots, x_\ell) = f(g_1(x_1, \ldots, x_\ell), \ldots, g_k(x_1, \ldots, x_\ell))\,. \tag{4}$$

Definition 12. *A* clone *of operations, C, is a set of operations on D that contains all projections and is closed under superposition. The k-ary operations in a clone C will be denoted by $C^{(k)}$.*

Example 3. For any D, let $\mathbf{J}_D$ be the set of all projections on D and $\mathcal{O}_D$ be the set of all operations on D. By Definition 12, both $\mathbf{J}_D$ and $\mathcal{O}_D$ are clones.

It is well known that $\mathrm{Pol}(\Gamma)$ is a clone for all valued constraint languages Γ [17].

Definition 13. *A k-ary* weighting *of a clone C is a function $\omega : C^{(k)} \to \mathbb{Q}$ such that $\omega(f) < 0$ only if f is a projection and*

$$\sum_{f \in C^{(k)}} \omega(f) \;=\; 0\,. \tag{5}$$

We define $\mathrm{supp}(\omega) = \{f \in C^{(k)} \mid \omega(f) > 0\}$.

Definition 14. *For any clone C, any k-ary weighting ω of C, and any $g_1, \ldots, g_k \in C^{(\ell)}$, the* superposition *of ω and $g_1, \ldots, g_k$, is the function $\omega[g_1, \ldots, g_k] : C^{(\ell)} \to \mathbb{Q}$ defined by*

$$\omega[g_1, \ldots, g_k](f') = \sum_{\{f \in C^{(k)} \mid f[g_1, \ldots, g_k] = f'\}} \omega(f)\,. \tag{6}$$

If ω satisfies (5) *then so does $\omega[g_1, \ldots, g_k]$. If the result of a superposition is a valid weighting (that is, negative weights are only assigned to projections) then that superposition will be called a* proper *superposition.*

We remark that the superposition (of an operation with other operations) is also known as composition. On the other hand, the superposition of a k-ary weighting ω with k ℓ-ary operations $g_1, \ldots, g_k$ can be seen as multiplying ω, seen as a (row) vector, by a matrix with rows indexed by k-ary operations and columns indexed by ℓ-ary operations; given a row operation f and a column operation f' the corresponding entry in the matrix is 1 if $f[g_1, \ldots, g_k] = f'$ and 0 otherwise. The result of this matrix multiplication is a vector of weights assigned to ℓ-ary operations.

Definition 15. *A* weighted clone*, W, is a non-empty set of weightings of some fixed clone C, called the* support clone *of W, which is closed under nonnegative scaling, addition of weightings of equal arity, and proper superposition with operations from C. We define* $\mathrm{supp}(W) = \bigcup_{\omega \in W} \mathrm{supp}(\omega)$.

Example 4. Let C be a clone. We give examples of two weighted clones with support clone C.

1. $\mathbf{W}_C^0$ is the zero-valued weighted clone, that is, the weighted clone containing, for each arity k, a weighting $\omega_k \in \mathbf{W}_C^0$ with $\omega_k(f) = 0$ for all $f \in C^{(k)}$.

2. $\mathbf{W}_C$ is the weighted clone containing all possible weightings of C.

By Definition 4, weighted clones are closed under nonnegative scaling. Consequently, by scaling by zero, any weighted clone W with support clone C contains $\mathbf{W}_C^0$, which is the inclusion-wise smallest weighted clone with support clone C. On the other hand, $\mathbf{W}_C$ is the inclusion-wise largest weighted clone with support clone C.

Example 5. It is easy to show that $\mathrm{supp}(W) \cup \mathbf{J}_D$ is a clone for any weighted clone W defined on D [10], see also [35, 47].

We now establish a correspondence between weightings and weighted relations, which will allow us to link weighted clones and weighted relational clones.

Definition 16. *Let ϕ be an m-ary weighted relation on D and let ω be a k-ary weighting of a clone C of operations on D. We call ω a* weighted polymorphism *of ϕ if $C \subseteq \mathrm{Pol}(\phi)$ and for any $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in D^m$ with $\mathbf{x}_i \in \mathrm{Feas}(\phi)$ for all $1 \leq i \leq k$, we have*

$$\sum_{f \in C^{(k)}} \omega(f)\phi(f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)) \;\leq\; 0\,. \tag{7}$$

If ω is a weighted polymorphism of ϕ we say that ϕ is improved *by ω.*

Example 6. Consider the class of submodular functions [37]. These are precisely the functions ϕ defined on $D = \{0, 1\}$ satisfying $\phi(\min(\mathbf{x}_1, \mathbf{x}_2)) + \phi(\max(\mathbf{x}_1, \mathbf{x}_2)) - \phi(\mathbf{x}_1) - \phi(\mathbf{x}_2) \leq 0$, where min and max are the two binary operations that return the smaller and larger of its two arguments respectively with respect to the usual order $0 < 1$. In other words, the set of submodular functions is the set of weighted relations with a binary weighted polymorphism ω_{sub} defined by: $\omega_{sub}(f) = -1$ if $f \in \{e_1^{(2)}, e_2^{(2)}\}$, $\omega_{sub}(f) = +1$ if $f \in \{\min, \max\}$, and $\omega_{sub}(f) = 0$ otherwise.

Definition 17. *For any $\Gamma \subseteq \mathbf{\Phi}_D$, we define* $\mathrm{wPol}(\Gamma)$ *to be the set of all weightings of* $\mathrm{Pol}(\Gamma)$ *which are weighted polymorphisms of all weighted relations $\phi \in \Gamma$. We write* $\mathrm{wPol}(\phi)$ *for* $\mathrm{wPol}(\{\phi\})$.

Definition 18. *We define $\mathbf{W}_D$ to be the union of the sets $\mathbf{W}_C$ over all clones C on D.*

Any $W \subseteq \mathbf{W}_D$ may contain weightings of *different* clones over D. We can then extend each of these weightings with zeros, as necessary, so that they are weightings of the same clone C, where C is the smallest clone containing all the clones associated with weightings in W.

Definition 19. *We define* $\mathrm{wClone}(W)$ *to be the smallest weighted clone containing this set of extended weightings obtained from W.*

For any $W \subseteq \mathbf{W}_D$, we denote by $\mathrm{Imp}(W)$ the set of all weighted relations in $\mathbf{\Phi}_D$ which are improved by all weightings $\omega \in W$.

Input	Mj	S1	S2	S3	P1	P2	P3	Mn
(x,x,y)	x	x	x	y	x	y	y	y
(x,y,x)	x	x	y	x	y	x	y	y
(y,x,x)	x	y	x	x	y	y	x	y

Table 1: Sharp ternary operations

Example 7. Every weighting in $\mathbf{W}^0_{\mathbf{J}_D}$ is a weighted polymorphism of any possible weighted relation. Hence $\mathrm{Imp}(\mathbf{W}^0_{\mathbf{J}_D}) = \mathbf{\Phi}_D$.

The weighted relations that are improved by all weightings are precisely those which take at most one value. Hence $\mathrm{Imp}(\mathbf{W}_{\mathbf{J}_D}) = \mathbf{R}_D$.

Definition 20. *A weighted clone W is called* tractable *if* $\mathrm{Imp}(W)$ *is tractable, and intractable if* $\mathrm{Imp}(W)$ *is intractable.*

The main result in [10] establishes a 1-to-1 correspondence between weighted relational clones and weighted clones.

Theorem 1 ([10])**.**

1. *For any finite D and any finite $\Gamma \subseteq \mathbf{\Phi}_D$,* $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma)$*.*

2. *For any finite D and any finite $W \subseteq \mathbf{W}_D$,* $\mathrm{wPol}(\mathrm{Imp}(W)) = \mathrm{wClone}(W)$*.*

Thus, when trying to identify tractable valued constraint languages, it is sufficient to consider only languages of the form $\mathrm{Imp}(W)$ for some weighted clone W.

Definition 21. *A weighting is called* positive *if it assigns positive weight to at least one operation that is not a projection.*

Positive weightings are necessary for tractability: any tractable weighted clone W contains a positive weighting [10, Corollary 7.4]. Consequently, throughout this paper we will be only concerned with weighted clones that contain a positive weighting.

2.4 Properties of operations

We finish this section with a discussion of certain types of operations. For any $k \geq 2$, a k-ary operation f is called *sharp* if f is not a projection, but the operation obtained by equating any two inputs in f is a projection [15]. All sharp operations must satisfy the identity $f(x, x, \dots, x) = x$; such operations are called *idempotent*. Ternary sharp operations may be classified according to their labels on tuples of the form $(x, x, y), (x, y, x)$ and (y, x, x), which must be equal to either x or y. There are precisely 8 possibilities, as listed in Table 1.

The first column in Table 1 corresponds to operations that satisfy the identities $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$ for all $x, y \in D$; such operations are called *majority* operations. The last column in the table corresponds to operations that satisfy the identities $f(x, x, y) = f(x, y, x) = f(y, x, x) = y$ for all $x, y \in D$; such operations are called *minority* operations. Columns 5, 6, and 7 in Table 1 correspond to operations that satisfy the identities $f(y, y, x) = f(x, y, x) = f(y, x, x) = y$ for all $x, y \in D$ (up to permutations of inputs); such operations are called *Pixley* operations [15]. For any $k \geq 3$, a k-ary operation f is called a *semiprojection* if it is not a projection, but there is an index $1 \leq i \leq k$ such that $f(x_1, \dots, x_k) = \mathrm{e}^{(k)}_i$ for all $x_1, \dots, x_k \in D$ such that $x_1, \dots, x_k$ are not

pairwise distinct. In other words, a semiprojection is a particular form of sharp operation where the operation obtained by equating any two inputs is always the *same* projection. Columns 2, 3, and 4 in Table 1 correspond to semiprojections.

It turns out that the only sharp operations of arity $k \geq 4$ are semiprojections. In other words, given an operation of arity ≥ 4, if every operation arising from the identification of two variables is a projection, then these projections coincide.

Lemma 1 (Świerczkowski's Lemma [41])**.** *The only sharp operations of arity $k \geq 4$ are semiprojections.*

We will need a technical lemma. But first we will introduce some notation. We denote by Cycl_k the set of k cyclic permutations on $\{1, \ldots, k\}$. We denote by $\circ$ the composition of two permutations, that is, for any $\sigma, \pi \in \mathrm{Cycl}_k$ we have $\sigma \circ \pi \in \mathrm{Cycl}_k$ is defined by $\sigma \circ \pi(x) = \sigma(\pi(x))$. For a k-ary operation f and a permutation $\pi \in \mathrm{Cycl}_k$ we will denote by f^π the operation $f^\pi = f[\mathrm{e}^{(k)}_{\pi(1)}, \ldots, \mathrm{e}^{(k)}_{\pi(k)}]$, that is, $f^\pi(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$.

Lemma 2. *Let W be a weighted clone and $\omega \in W$ a positive k-ary weighting. Then there is a positive k-ary weighting $\mu \in W$ with the following properties:*

1. $\mathrm{supp}(\mu) = \bigcup_{f \in \mathrm{supp}(\omega)} \bigcup_{\pi \in \mathrm{Cycl}_k} f^\pi$*;*
2. $\mu(\mathrm{e}^{(k)}_i) = -1$ *for every* $1 \leq i \leq k$*;*
3. $\mu(f) = \mu(f^\pi)$ *for every* $f \in \mathrm{supp}(\mu)$ *and* $\pi \in \mathrm{Cycl}_k$.

Proof. Let

$$\omega' \;=\; \sum_{\pi \in \mathrm{Cycl}_k} \omega[\mathrm{e}^{(k)}_{\pi(1)}, \ldots, \mathrm{e}^{(k)}_{\pi(k)}]\,. \tag{8}$$

Let $f \in \mathrm{supp}(\omega)$ and $\pi \in \mathrm{Cycl}_k$. We have

$$\begin{aligned}\omega'(f) \;&=\; \sum_{g \in \mathrm{supp}(\omega)} \sum_{\substack{\sigma \,\in\, \mathrm{Cycl}_k \\ g^\sigma \,=\, f}} \omega(g) \;=\; \sum_{g \in \mathrm{supp}(\omega)} \sum_{\substack{\sigma \circ \pi \,\in\, \mathrm{Cycl}_k \\ g^{\sigma \circ \pi} \,=\, f^\pi}} \omega(g) \;= \\ &=\; \sum_{g \in \mathrm{supp}(\omega)} \sum_{\substack{\sigma' \,\in\, \mathrm{Cycl}_k \\ g^{\sigma'} \,=\, f^\pi}} \omega(g) \;=\; \omega'(f^\pi)\,.\end{aligned} \tag{9}$$

Thus ω' satisfies the first and the third property of the lemma.

Since ω is positive we have that $\sum_{i=1}^{k} \omega(\mathrm{e}^{(k)}_i) < 0$ and thus, by (9), we have $\omega'(\mathrm{e}^{(k)}_i) < 0$ for every $1 \leq i \leq k$. Let $\omega'(\mathrm{e}^{(k)}_1) = w$. By (9) again, $\omega'(\mathrm{e}^{(k)}_i) = w$ for every $1 \leq i \leq k$. Thus $\mu = \frac{1}{w}\omega'$ satisfies all three properties of the lemma. ☐

2.5 Cores

We show that with respect to tractability, the only interesting weighted clones (and thus weighted relational clones) are those whose unary weightings can assign positive weight only to very special operations.

The idea of cores and rigid cores originated in the algebraic approach to CSPs [7, 25] and has also proved useful in the complexity classification of finite-valued CSPs [23, 46].

Definition 22. *A weighted clone W is a* core *if for every unary weighting $\omega \in W$ every operation $f \in \operatorname{supp}(\omega)$ is bijective. A valued constraint language Γ is a core if $W = \operatorname{wPol}(\Gamma)$ is a core.*

Theorem 2. *Let Γ be a valued constraint language on D. If Γ is not a core then there is a core valued constraint language Γ' on $D' \subseteq D$ such that Γ is tractable if and only if Γ' is tractable and Γ is intractable if and only if Γ' is intractable.*

Proof. Let $\omega \in \operatorname{wPol}(\Gamma)$ be a positive unary weighting. By scaling by $1/|\omega(\mathrm{e}_1^{(1)})|$, we have $\omega(\mathrm{e}_1^{(1)}) = -1$ and thus $\sum_{f \in \operatorname{supp}(\omega)} \omega(f) = 1$. For any weighted relation $\phi \in \Gamma$ of arity m and any m-tuple $\mathbf{x} \in D^m$, we have $(*)$ $\phi(\mathbf{x}) \geq \sum_{f \in \operatorname{supp}(\omega)} \omega(f)\phi(f(\mathbf{x}))$. Suppose that $\mathbf{y}$ is a minimal-cost assignment for ϕ; that is, $\phi(\mathbf{y}) \leq \phi(\mathbf{x})$ for all $\mathbf{x} \in D^m$. Then for every $f \in \operatorname{supp}(\omega)$, we have $f(\mathbf{y})$ is also a minimal-cost assignment. Assume for contrary that for some $f' \in \operatorname{supp}(\omega)$, we have $\phi(f'(\mathbf{y})) > \phi(\mathbf{y})$; write $\phi(f'(\mathbf{y})) = \phi(\mathbf{y}) + \epsilon$, where $\epsilon > 0$. Then we claim that there is an $f \in \operatorname{supp}(\omega)$ such that $\phi(f(\mathbf{y})) < \phi(\mathbf{y})$, which contradicts the choice of $\mathbf{y}$. To prove the claim assume that $\phi(f(\mathbf{y})) \geq \phi(\mathbf{y})$ for every $f \in \operatorname{supp}(\omega) \setminus \{f'\}$. Hence we get $\sum_{f \in \operatorname{supp}(\omega)} \omega(f)\phi(f(\mathbf{y})) = \sum_{f \in \operatorname{supp}(\omega)\setminus\{f'\}} \omega(f)\phi(f(\mathbf{y})) + \omega(f')\phi(f'(\mathbf{y})) \geq (1 - \omega(f')\phi(\mathbf{y}) + \omega(f')(\phi(\mathbf{y}) + \epsilon) = \phi(\mathbf{y}) + \omega(f')\epsilon > \phi(\mathbf{y})$, which contradicts $(*)$.

Consequently, given an instance $I \in \operatorname{VCSP}(\Gamma)$ and a solution s to I, we can take any unary weighting $\omega \in \operatorname{wPol}(\Gamma)$ and any unary operation $f \in \operatorname{supp}(\omega)$ and get another solution $f(s)$ to I; the solution $f(s)$ uses only labels from $f(D)$. Consider a unary non-bijective operation $f \in \operatorname{supp}(\omega)$ with the minimum $|f(D)|$ over all unary weightings $\omega \in \operatorname{wPol}(\Gamma)$. We denote by $D' = f(D)$ the range of f. We denote by Γ' language containing the restriction of every $\phi \in \Gamma$ to D'.

Given any instance $I \in \operatorname{VCSP}(\Gamma)$ we can create, by replacing each weighted relation ϕ in I by ϕ', in polynomial time an instance $I' \in \operatorname{VCSP}(\Gamma')$ with the following properties: any solution to I' is also a solution to I, and for any solution s to I we have that $f(s)$ is a solution to I'. If Γ' is not a core we can repeat the same construction with Γ'. $\square$

Theorem 2 was independently obtained in [35], where it was also shown that, with respect to tractability, it suffices to restrict to rigid cores.

Definition 23. *A weighted clone W is a* rigid core *if the only unary operation in the support clone of W is the unary projection $\mathrm{e}_1^{(1)}$. A valued constraint language Γ is a rigid core if $W = \operatorname{wPol}(\Gamma)$ is a rigid core; that is if the only unary polymorphism of Γ is $\mathrm{e}_1^{(1)}$.*

Theorem 3 ([35])**.** *Let Γ be a valued constraint language on D. If Γ is not a rigid core then there is a rigid core valued constraint language Γ' on $D' \subseteq D$ such that Γ is tractable if and only if Γ' is tractable and Γ is intractable and only if Γ' is intractable.*

It is not hard to show that a weighted clone W (a valued constraint language Γ) is a rigid core if and only if all operations in the support clone of W (polymorphisms of Γ, respectively) are idempotent.

3 Conditions for Tractability

In this section we will present our main results.

Creed and Živný obtained the following result on the structure of weighted clones with a positive weighting [13, Theorem 2]; see also [10, Corollary 7.7].

Theorem 4 ([13])**.** *Any weighted clone W containing a positive weighting contains a weighting whose support is either:*

1. *a set of unary operations that are not projections; or*

2. *a set of binary idempotent operations that are not projections; or*

3. *a set of ternary operations that are majority operations, minority operations, Pixley operations or semiprojections; or*

4. *a set of k-ary semiprojections (for some $k > 3$).*

Since rigid cores require all unary weightings be zero-valued, case (1) of Theorem 4 can be easily eliminated. Moreover, using Gordan's Theorem (a variant of Farkas' Lemma) we will strengthen Theorem 4 by refining the ternary case, thus obtaining the following result, which is the main result of this paper.

Theorem 5 (**Main**). *Any weighted clone W that is a rigid core and contains a positive weighting also contains a weighting whose support is either:*

1. *a set of binary idempotent operations that are not projections; or*

2. *a set of ternary operations that are either:*

   (a) *a set of majority operations; or*

   (b) *a set of minority operations; or*

   (c) *a set of majority operations with total weight 2 and a set of minority operations with total weight 1; or*

3. *a set of k-ary semiprojections (for some $k \geq 3$).*

The proof of Theorem 5 can be found in Section 4.

Note that compared to Theorem 4 the inequality in case (3) of Theorem 5 is not strict as it includes one of the ternary cases.

We remark that Theorem 5 holds for *any* weighted clone W with *any* support clone C as long as W contains a positive weighting.

Theorem 5 tells us that (i) Pixley operations are not necessary for tractability, (ii) semiprojections can be separated from the other types of ternary operations, and (iii) the only possible interplay between majority and minority operations, as described in case (2c) of Theorem 5.

We now focus on the weighted clones containing one of the weightings described in Theorem 5.

Case (1) of Theorem 5 A weighting described in Theorem 5 (1) can lead both to tractable and intractable weighted clones, as the next two examples demonstrate, but the precise boundary of tractability is currently unknown.

Example 8. A binary operation $f : D^2 \to D$ is called conservative if $f(x, y) \in \{x, y\}$ for all $x, y \in D$ and commutative if $f(x, y) = f(y, x)$ for all $x, y \in D$. Moreover, f is called a *tournament* operation if f is both conservative and commutative. Let W be a weighted clone such $\mathrm{supp}(W)$ contains a tournament operation. Then, by a recent result of the authors [47], W is tractable.

Example 9. Let $D = \{0, 1, 2, 3\}$ and let f be a binary operation defined by Table 2. Note that f is an idempotent operation but not a projection. In fact, f is an example of a *rectangular band* [36], which is an idempotent and associative binary operation $f : D^2 \to D$ satisfying $f(x, f(y, z)) = f(x, z)$ for all $x, y, z \in D$. Let W be the weighted clone generated by the weighting ω defined by $\omega(\mathrm{e}_1^{(2)}) = \omega(\mathrm{e}_2^{(2)}) = -1$ and $\omega(f) = +2$. It is known that W is intractable [26, 38].

f	**0**	**1**	**2**	**3**
0	0	1	0	1
1	0	1	0	1
2	2	3	2	3
3	2	3	2	3

Table 2: Definition of f from Example 9.

Case (2a) of Theorem 5 A weighting as described in Theorem 5 (2a) implies tractable weighted clones, as we will now show.

A weighted relational clone that contains only relations (and thus is a relational clone) is called *crisp*. A weighted clone W is called crisp if $\mathrm{Imp}(W)$ is a crisp weighted relational clone.

Proposition 6. *Let W be a weighted clone with a positive ternary weighting $\omega \in W$ such that all operations $f \in \mathrm{supp}(\omega)$ are majority operations. Then W is crisp.*

In order to prove Proposition 6, we prove a more general result. A k-ary operation $f : D^k \to D$, where $k \geq 3$, is called a *near-unanimity* operation if for all $x, y \in D$,

$$f(y, x, x, \ldots, x) = f(x, y, x, x, \ldots, x) = \cdots = f(x, x, \ldots, x, y) = x\,. \tag{10}$$

Note that a ternary near-unanimity operation is a majority operation.

Proposition 7. *Let W be a weighted clone with a positive weighting $\omega \in W$ such that all operations $f \in \mathrm{supp}(\omega)$ are near-unanimity operations. Then W is crisp.*

Proof. Let ω be k-ary. Note that if f is a k-ary near-unanimity operation then so is $g(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$ for any permutation π on $\{1, \ldots, k\}$. Thus, by Lemma 2, we can assume ω assigns weight -1 to each of the k projections (and still every $f \in \mathrm{supp}(\omega)$ is a near-unanimity operation).

Let $\phi \in \mathrm{Imp}(W)$ be an m-ary weighted relation and let $\mathbf{x}, \mathbf{y} \in D^m$ be such that $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\phi)$. Since $\omega \in \mathrm{wPol}(\phi)$, we have, by (7) with $\mathbf{x}_1 = \mathbf{y}$ and $\mathbf{x}_i = \mathbf{x}$ for all $2 \leq i \leq k$, and by (10), $-\phi(\mathbf{y}) - (k-1)\phi(\mathbf{x}) + k\phi(\mathbf{x}) \leq 0$, which gives $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$. By swapping $\mathbf{x}$ and $\mathbf{y}$ in (7), we get $-\phi(\mathbf{x}) - (k-1)\phi(\mathbf{y}) + k\phi(\mathbf{y}) \leq 0$, which gives $\phi(\mathbf{y}) \leq \phi(\mathbf{x})$. Together, $\phi(\mathbf{x}) = \phi(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\phi)$. □

Since crisp weighted relational clones with a near-unanimity polymorphism are tractable [26], we get the following.

Corollary 1. *A weighted clone containing a positive weighting ω with all operations in $\mathrm{supp}(\omega)$ being near-unanimity operations is tractable.*

Case (2b) of Theorem 5 A weighting as described in Theorem 5 (2b) also implies tractable weighted clones, as we will now show.

Proposition 8. *Let W be a weighted clone with a positive ternary weighting $\omega \in W$ such that all operations $f \in \mathrm{supp}(\omega)$ are minority operations. Then W is crisp.*

Proof. Note that if f is a minority operation then so is $g(x_1, x_2, x_3) = f(x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)})$ for any permutation π on $\{1, 2, 3\}$. Thus, by Lemma 2, we can assume ω assigns weight -1 to each of the three projections (and still every $f \in \text{supp}(\omega)$ is a minority operation).

Let $\phi \in \text{Imp}(W)$ be an m-ary weighted relation and let $\mathbf{x}, \mathbf{y} \in D^m$ be such that $\mathbf{x}, \mathbf{y} \in \text{Feas}(\phi)$. Since $\omega \in \text{wPol}(\phi)$, we have, by (7) with $\mathbf{x}_1 = \mathbf{x}$ and $\mathbf{x}_2 = \mathbf{x}_3 = \mathbf{y}$, $-\phi(\mathbf{x}) - 2\phi(\mathbf{y}) + 3\phi(\mathbf{x}) \leq 0$, which gives $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$. By swapping $\mathbf{x}$ and $\mathbf{y}$ in (7), we get $-\phi(\mathbf{y}) - 2\phi(\mathbf{x}) + 3\phi(\mathbf{y}) \leq 0$, which gives $\phi(\mathbf{y}) \leq \phi(\mathbf{x})$. Together, $\phi(\mathbf{x}) = \phi(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \text{Feas}(\phi)$. ☐

Since crisp weighted relational clones with a minority polymorphism are tractable [26], we get the following.

Corollary 2. *A weighted clone containing a positive weighting ω with all operations in* $\text{supp}(\omega)$ *being minority operations is tractable.*

Case (2c) of Theorem 5 In a recent paper the authors have shown [47] that any weighting described in Theorem 5 (2c) implies tractability. This is a corollary of the following result.

Theorem 9 ([47]). *Let W be a weighted clone. If there is a weighting $\omega \in W$ such that* $\text{supp}(\omega)$ *contains a majority operation then W is tractable.*

Previously, only a special type of the weightings described in Theorem 5 (2c) has been known to imply tractability.

Example 10. A k-ary weighting ω is a *multimorphism* if $\omega(f) \in \mathbb{N}$ for all $f \in \text{supp}(\omega)$ and $\omega(\mathrm{e}_i^{(k)}) = -1$ for all $1 \leq i \leq k$ [12]. It has been shown that if a weighted clone W contains a weighting ω described in Theorem 5 (2c) such that ω is a multimorphism then W is tractable [34].

Case (3) of Theorem 5 We show that the weightings described in Theorem 5 (3) *alone* are not sufficient for tractability. As in case (1), the precise boundary of tractability is currently unknown.

Example 11. Let D be a fixed set with $|D| > 2$. Fix two distinct labels from D, say $0, 1 \in D$. Let ϕ be the following ternary weighted relation: $\phi(x, y, z) = \infty$ if $\{x, y, z\} = \{0\}$, or $\{x, y, z\} = \{1\}$, or $\{x, y, z\} \neq \{0, 1\}$; and $\phi(x, y, z) = 0$ otherwise. The weighted relation ϕ corresponds to the Not-All-Equal Satisfiability problem [19], which is NP-hard [40]. It is easy to show that every semiprojection on D is a polymorphism of ϕ. Take a k-ary semiprojection f for some $k \geq 3$ and $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \text{Feas}(\phi)$. From the definition of ϕ, we have $\mathbf{x}_i \in \{0, 1\}^3$ for every $1 \leq i \leq k$. Since there are at most two distinct labels in each coordinate, $f(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ reduces to a projection (from the definition of semiprojections) and thus f is a polymorphism of ϕ as $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \mathbf{x}_i$ for some $1 \leq i \leq k$.

Let C be the clone of operations on D generated by all semiprojections on D. Let $W = \mathbf{W}_C$ be the weighted clone containing all possible weightings of C. In particular, W contains all possible weightings whose support contains only semiprojections. Since $C \subseteq \text{Pol}(\phi)$ and ϕ is a relation we have that $\phi \in \text{Imp}(W)$. Consequently, W is intractable.

Finite-Valued Weighted Clones Recall that valued constraint languages capture both decision and optimisation problems. Clones, which capture crisp valued constraint languages and thus purely decision problems, have been studied extensively in universal algebra [22, 42]. We now focus on an important special type of weighted clones that correspond to valued constraint languages that capture purely optimisation problems. Such valued constraint languages are called *finite-valued* as they only contain finite-valued weighted relations.

Weighted clones corresponding to finite-valued constraint languages (together with the binary equality relation $\phi_=$) are those with support clone $\mathcal{O}_D$. To see this, we denote, for a clone C, by $\mathrm{Inv}(C)$ the relational clone that consists of relations R with $f \in \mathrm{Pol}(R)$ for every $f \in C$. Then, it is well known that $\mathrm{Inv}(\mathcal{O}_D) = \mathrm{RelClone}(\{\phi_=\})$ and observe that $\mathrm{Feas}(\mathrm{Imp}(W)) \subseteq \mathrm{Inv}(\mathcal{O}_D)$ for any weighted clone W with support clone $\mathcal{O}_D$.[4]

However, as we have limited our scope to rigid cores (which, by Theorem 3, does not change tractability), we will define a weighted clone W to be finite-valued if its support clone is equal to $\mathbf{I}_D$, the clone of all *idempotent* operations on D.

Definition 24. *A weighted clone W on D is called* finite-valued *if the support clone of W is $\mathbf{I}_D$.*

For any $d \in D$, the unary constant relation ϕ_d is defined by $\phi_d(x) = 0$ if $x = d$ and $\phi_d(x) = \infty$ otherwise. Let $\mathcal{R} = \mathrm{RelClone}(\{\phi_=\} \cup \cup_{d\in D}\{\phi_d\})$. It is known that $\mathrm{Inv}(\mathbf{I}_D) = \mathcal{R}$ [7].

The weighted relational clones corresponding to finite-valued weighted clones are those that are subsets of the weighted relational clone generated by $\mathcal{R}$ and finite-valued weighted relations.

We already know that weighted clones containing any of the weightings described in Theorem 5 (2a-c) are tractable. In fact, in the finite-valued case, the corresponding weighted relational clones are rather trivial as we will now show.

Let W be a finite-valued weighted clone on D. Then for any weighted relation $\phi \in \mathrm{Imp}(W)$ we have $\mathrm{Feas}(\phi) \in \mathcal{R}$.

If W contains a weighting described in Theorem 5 (2a) then, by Proposition 6, $\mathrm{Imp}(W)$ is crisp and thus every $\phi \in \mathrm{Imp}(W)$ can be written as the addition of a rational constant to a weighted relation in $\mathcal{R}$. Hence $\mathrm{Imp}(W)$ is tractable. Similarly, if W contains a weighting described in Theorem 5 (2b) then, by Proposition 8, $\mathrm{Imp}(W)$ is crisp and thus every $\phi \in \mathrm{Imp}(W)$ can be written as the addition of a rational constant to a weighted relation in $\mathcal{R}$. Hence $\mathrm{Imp}(W)$ is tractable.

The next result shows that a weighting described in Theorem 5 (2c) also suffices for tractability in the finite-valued case.

Proposition 10. *Let W be a finite-valued weighted clone. If W contains a positive weighting described in Theorem 5 (2c) then every weighted relation $\phi \in \mathrm{Imp}(W)$ can be expressed as a sum of unary weighted relations and the binary equality relation $\phi_=$.*

Proof. By Lemma 2, we can assume the weighting assigns weight -1 to each of the three projections and still is as described in Theorem 5 (2c).

An m-ary relation R on D is called trivial if $R = D^m$. First we show than any relation $R \in \mathcal{R}$ can be expressed as a sum of unary relations, trivial relations, and $\phi_=$. The claim holds true for the generators of $\mathcal{R}$, that is, for $\phi_=$ and ϕ_d for all $d \in D$. Next, if $R = R_1 \wedge R_2$ and the claim holds true for both R_1 and R_2 the it also holds true for R. Finally, let $R = \exists x R'$ and assume that R' satisfies the claim. If x appears in some $\phi_=$ in R', say $\phi_=(x, x')$, then we can replace all occurrences of x by x'. Otherwise, x appears only in constant and trivial relations in R'. If the conjunction of the unary relations that x appears in is empty then the claim holds trivially. Otherwise, we can replace x by any other variable.

Consequently, for any $\phi \in \mathrm{Imp}(W)$, $\mathrm{Feas}(\phi)$ can be written as a sum of unary relations, trivial relations, and $\phi_=$. Observe that any ϕ with $\mathrm{Feas}(\phi) = \phi_=$ can be written as a sum of $\phi_=$ and the unary weighted relation $\phi'(x) = \phi(x, x)$. Thus, it remains to show that any $\phi \in \mathrm{Imp}(W)$ with $\mathrm{Feas}(\phi)$ being a trivial relation can be written as a sum of unary weighted relations.

[4] More generally, we have $\mathrm{Feas}(\mathrm{Imp}(W)) = \mathrm{Inv}(C)$ for any nonempty weighted clone W with support clone C. On the one hand, if $\phi \in \mathrm{Imp}(W)$ then, by Definition 16, $C \subseteq \mathrm{Pol}(\phi)$, which implies $\mathrm{Feas}(\phi) \in \mathrm{Inv}(\mathrm{Pol}(\phi)) \subseteq \mathrm{Inv}(C)$. On the other hand, if $R \in \mathrm{Inv}(C)$ then $C \subseteq \mathrm{Pol}(R)$. Since R satisfies (7) we have $R \in \mathrm{Imp}(W)$.

For any m-tuple $\mathbf{x} \in D^m$, we will write $\mathbf{x}[i \leftarrow d]$ to denote the tuple with $d \in D$ substituted at position i. In other words, $\mathbf{x}[i \leftarrow d]$ is the m-tuple identical to $\mathbf{x}$ except (possibly) at position i, where it is equal to d.

We will use [12, Lemma 6.23] which says that a weighted relation $\phi : D^m \to \mathbb{Q}$ can be expressed as a sum of unary weighted relations if and only if, for all $\mathbf{x}, \mathbf{y} \in D^m$ and all $1 \leq i \leq m$, we have

$$\phi(\mathbf{x}) + \phi(\mathbf{y}) = \phi(\mathbf{x}[i \leftarrow \mathbf{y}[i]]) + \phi(\mathbf{y}[i \leftarrow \mathbf{x}[i]])\,. \tag{11}$$

Take any $\mathbf{x}, \mathbf{y} \in D^m$ and $1 \leq i \leq m$. Let $a = \mathbf{x}[i]$ and $b = \mathbf{y}[i]$. Now consider the tuples $\mathbf{x}$, $\mathbf{x}[i \leftarrow b]$, and $\mathbf{y}[i \leftarrow a]$. By applying the weighting from the statement of the proposition as in (7), we get $\phi(\mathbf{x})+\phi(\mathbf{x}[i \leftarrow b])+\phi(\mathbf{y}[i \leftarrow a]) \geq 2\phi(\mathbf{x})+\phi(\mathbf{y})$ and thus $\phi(\mathbf{x}[i \leftarrow b])+\phi(\mathbf{y}[i \leftarrow a]) \geq \phi(\mathbf{x})+\phi(\mathbf{y})$. Now consider the tuples $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{y}[i \leftarrow a]$. By applying the weighting from the statement of the proposition as in (7), we get $\phi(\mathbf{x}) + \phi(\mathbf{y}) + \phi(\mathbf{y}[i \leftarrow a]) \geq 2\phi(\mathbf{y}[i \leftarrow a]) + \phi(\mathbf{x}[i \leftarrow b])$ and thus $\phi(\mathbf{x}) + \phi(\mathbf{y}) \geq \phi(\mathbf{y}[i \leftarrow a]) + \phi(\mathbf{x}[i \leftarrow b])$. □

Corollary 3. *A finite-valued weighted clone containing a positive weighting ω described in Theorem 5 (2c) is tractable.*

The only remaining finite-valued weighted clones contain a weighting that is either as described in Theorem 5 (1) or as described in Theorem 5 (3). We have seen an example of a (tractable) weighted clone with a weighting as described in Theorem 5 (1) in Example 8.

We now give an example of an *intractable finite-valued* weighted clone with a weighting as described in Theorem 5 (1). (We note that the intractability of the weighted clone W from Example 9, which contains a weighting as described in Theorem 5 (1), relies on the fact that W is not finite-valued and thus is not immediately applicable here.)

Example 12. Let $D = \{0, 1, 2\}$. Recall from Example 8 that a binary operation $f : D^2 \to D$ is *conservative* if $f(x, y) \in \{x, y\}$ for all $x, y \in D$. For any conservative binary operation $f : D^2 \to D$ and any 2-element subdomain $\{a, b\} \subseteq D$, the restriction $f|_{\{a,b\}}$ of f onto $\{a, b\}$ behaves either as $\mathrm{e}_1^{(2)}$, $\mathrm{e}_2^{(2)}$, min, or max, where min and max are the two operations that return the smaller (larger) of its two arguments with respect to the usual order $0 < 1 < 2$, respectively. Consider the operations in Table 3 described by their behaviour on the various 2-element subdomains.

f	**{0,1}**	**{0,2}**	**{1,2}**	$\boldsymbol{\omega}(\mathbf{f})$
f_1	$\mathrm{e}_1^{(2)}$	$\mathrm{e}_1^{(2)}$	$\mathrm{e}_1^{(2)}$	-0.5
f_2	$\mathrm{e}_2^{(2)}$	$\mathrm{e}_2^{(2)}$	$\mathrm{e}_2^{(2)}$	-0.5
f_3	$\mathrm{e}_1^{(2)}$	min	$\mathrm{e}_1^{(2)}$	0.5
f_4	$\mathrm{e}_2^{(2)}$	min	$\mathrm{e}_2^{(2)}$	0.5

Table 3: Definition of ω

Note that $f_1 = \mathrm{e}_1^{(2)}$ and $f_2 = \mathrm{e}_2^{(2)}$. The weighting ω is defined by the last column of Table 3. Note that ω is not commutative. It can be checked that ω is a weighted polymorphism of the finite-valued weighted relation $\phi : \{0, 1, 2\} \to \mathbb{Q}$ defined in Table 4.

Now since $\operatorname{argmin} \phi = \{(0, 1), (1, 0)\}$, we have that ϕ satisfies the (MC) condition [46] and thus can be used to reduce from Max-Cut [12]. Thus, $W = \mathrm{wClone}(\{\omega\})$ is intractable.

Thus weightings described in Theorem 5 (1) can lead to both tractable and intractable finite-valued weighted clones. The authors have recently shown that with respect to tractability of finite-valued constraint languages the necessary and sufficient condition is having a binary weighting ω

ϕ	**0**	**1**	**2**
0	1	0	1
1	0	1	1
2	1	1	1

Table 4: Definition of ϕ

that assigns positive weight to idempotent *commutative* operations only; that is, for every $f \in \text{supp}(\omega)$ we have $f(x, y) = f(y, x)$ for all $x, y \in D$ [46].[5] However, the precise interplay of case (1) and case (3) of Theorem 5 is currently unknown.

Minimal Weighted Clones Any weighted relational clone $\Gamma \subseteq \mathbf{\Phi}_D$ with $\text{wRelClone}(\Gamma) = \mathbf{\Phi}_D$ is NP-hard. A weighted relational clone on D is called *maximal* if it is as large as possible but $\text{wRelClone}(\Gamma) \neq \mathbf{\Phi}_D$.[6]

Definition 25. *A weighted relational clone $\Gamma \subseteq \mathbf{\Phi}_D$ is called* maximal *if* $\text{wRelClone}(\Gamma) \neq \mathbf{\Phi}_D$ *but for any $\phi \notin \Gamma$ we have* $\text{wRelClone}(\Gamma \cup \{\phi\}) = \mathbf{\Phi}_D$.

It follows that a valued constraint language Γ is maximal if and only if the weighted relational clone $\text{wRelClone}(\Gamma)$ is maximal.

As a special case of Definition 25, we get that a relational clone Γ is maximal if $\Gamma \neq \mathbf{R}_D$ but for any $R \in \mathbf{R}_D$ we have $\text{RelClone}(\Gamma \cup \{R\}) = \mathbf{R}_D$.

A weighted clone is called *minimal* if it is not zero-valued but the only weighted clone properly included in it is the zero-valued weighted clone.

Definition 26. *A weighted clone W with support clone C is called* minimal *if $W \neq \mathbf{W}^0_C$ and every positive weighting $\omega \in W$ satisfies* $\text{wClone}(\omega) = W$.

Maximal weighted relational clones correspond, via the Galois correspondence given in Theorem 1, to minimal weighted clones.

We will be interested in maximal *tractable* weighted relational clones and thus minimal tractable weighted clones. Maximal crisp weighted relational clones have been classified with respect to tractability in [5, 9]. We now show that that there are no tractable maximal non-crisp weighted relational clones.

Theorem 11. *All maximal non-crisp weighted relational clones are intractable.*

Proof. If Γ contains all finite-valued weighted relations then it is intractable. Otherwise, there is a finite-valued weighted relation $\phi \notin \Gamma$. Since ϕ is finite-valued we have $\text{Feas}(\Gamma) = \text{Feas}(\text{wRelClone}(\Gamma \cup \{\phi\}))$. But then either $\text{Feas}(\Gamma) = \mathbf{R}_D$, in which case Γ is intractable, or $\text{Feas}(\Gamma) = \text{Feas}(\text{wRelClone}(\Gamma \cup \{\phi\})) \neq \mathbf{R}_D$, in which case Γ is not maximal. $\square$

[5]The result from [46] extends from finite-valued constraint languages to finite-valued weighted relational clones as adding the binary equality relation and unary constant relations does not affect tractability in the presence of a binary commutative weighted polymorphism.

[6]A (tractable) valued constraint language Γ is called maximal in [12] if for any $\phi \notin \Gamma$, $\Gamma \cup \{\phi\}$ is intractable. We require $\text{wRelClone}(\Gamma \cup \{\phi\}) = \mathbf{\Phi}_D$, which implies the intractability of $\Gamma \cup \{\phi\}$, thus borrowing the concept of maximality from [4, 5, 9, 27] and extending it from relational clones to weighted relational clones.

4 Proof of Theorem 5

In this section we will prove the following theorem, which is our main result.

Theorem (Theorem 5 restated)**.** *Any weighted clone W that is a rigid core and contains a positive weighting also contains a weighting whose support is either:*

1. *a set of binary idempotent operations that are not projections; or*
2. *a set of ternary operations that are either:*
   (a) *a set of majority operations; or*
   (b) *a set of minority operations; or*
   (c) *a set of majority operations with total weight 2 and a set of minority operations with total weight 1; or*
3. *a set of k-ary semiprojections (for some $k \geq 3$).*

We will use the following variant of Farkas' Lemma.

Theorem 12 (Gordan)**.** *Let $A \in \mathbb{Q}^{n\times m}$ be a matrix. Either $Ax = 0$, where $x \in \mathbb{Q}^m$ with $x \geq 0$ and $x \neq 0$, or $\exists y \in \mathbb{Q}^n$ with $y^\intercal A > 0$.*

By Definition 15, only *proper* superpositions are allowed within a weighted clone. However, the following result from [10] shows that any weighted sum of arbitrary superpositions of a pair of weightings ω_1 and ω_2 can be obtained by taking a weighted sum of superpositions of ω_1 and ω_2 with projection operations, and then taking a superposition of the result. Given that superpositions with projections are always proper [10], this result implies that any weighting which can be expressed as a weighted sum of arbitrary (i.e., possibly improper) superpositions can also be expressed as a superposition of a weighted sum of *proper* superpositions.

Lemma 3 ([10, Lemma 6.4])**.** *Let C be a clone, and let ω_1 and ω_2 be weightings of C, of arity k and ℓ respectively. For any $g_1, \ldots, g_k \in C^{(m)}$ and any $g'_1, \ldots, g'_\ell \in C^{(m)}$,*

$$c_1\, \omega_1[g_1, \ldots, g_k] + \ c_2\, \omega_2[g'_1, \ldots, g'_\ell] = \omega[g_1, \ldots, g_k, g'_1, \ldots, g'_\ell]\,,$$

where $\omega = c_1\, \omega_1[e_1^{(k+\ell)}, \ldots, e_k^{(k+\ell)}] + \ c_2\, \omega_2[e_{k+1}^{(k+\ell)}, \ldots, e_{k+\ell}^{(k+\ell)}]$

Before proving Theorem 5 we introduce the following useful notion. For the reader's convenience, we repeat here Table 1 from Section 2. We call (ternary) operations corresponding to columns 5, 6, and 7 in Table 1 Pixley operations of type 1, 2, and 3 respectively, and will denote by P1 (P2 and P3) the Pixley operations of type 1 (2 and 3, respectively). We call (ternary) semiprojections corresponding to columns 2, 3, and 4 in Table 1 semiprojections of type 1, 2, and 3 respectively, and

Input	Mj	S1	S2	S3	P1	P2	P3	Mn
(x,x,y)	x	x	x	y	x	y	y	y
(x,y,x)	x	x	y	x	y	x	y	y
(y,x,x)	x	y	x	x	y	y	x	y

Table 1 (restated): Sharp ternary operations

will denote by S1 (S2 and S3) the semiprojections of type 1 (2 and 3, respectively). More generally, a k-ary semiprojection f is called of type $1 \leq i \leq k$ if equating any two inputs of f results in $\mathrm{e}_i^{(k)}$.

For any Pixley operation f of type $i \in \{1,2,3\}$ we can obtain, by (cyclically) permuting the arguments of f, Pixley operations of the other two types. For instance, if $f \in$ P1 then we have $g \in$ P2 and $h \in$ P3, where $g(x,y,z) = f[\mathrm{e}_3^{(3)}, \mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}] = f(z,x,y)$ and $h(x,y,z) = f[\mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}, \mathrm{e}_1^{(3)}] = f(y,z,x)$. Two Pixley operations f and g of different types are called *related* if there is a permutation $\pi \in \mathrm{Cycl}_3$ such that $f = g^\pi$. (Note that the requirement of f and g being of different types excludes the identity permutation $(1,2,3)$ and there are only other two permutations in Cycl_3, namely $(2,3,1)$ and $(3,1,2)$.)

Similarly, two semiprojections f and g of different types are called related if there is a permutation $\pi \in \mathrm{Cycl}_3$ such that $f = g^\pi$.

The following table, which can be verified using the definitions above, will be useful in the proof of Theorem 5. It lists the types of ternary sharp operations obtained by superposing a ternary sharp operation of an arbitrary type (columns in Table 5) with any of the three cyclic permutations of the three ternary projections (rows in Table 5).

Permutation	Mj	S1	S2	S3	P1	P2	P3	Mn
$\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}$	Mj	S1	S2	S3	P1	P2	P3	Mn
$\mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}, \mathrm{e}_1^{(3)}$	Mj	S2	S3	S1	P3	P1	P2	Mn
$\mathrm{e}_3^{(3)}, \mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}$	Mj	S3	S1	S2	P2	P3	P1	Mn

Table 5: Types of ternary sharp operations superposed with cyclic permutations of projections

Note that taking a semiprojection f of type i and a Pixley operation g of type i, f^π and g^π can be of different types; e.g., if f is a semiprojection of type 1 and g is a Pixley operation of type 1 and $\pi = (2,3,1)$ then f^π is a semiprojection of type 2 and g^π is a Pixley operation of type 3.

Proof of Theorem 5. It suffices to consider the ternary case as the rest of the theorem follows from (the proof of) Theorem 4 and the fact that W is a rigid core, which eliminates the first case of Theorem 4.

Let W be a weighted clone containing a ternary positive weighting ω such that every $f \in \mathrm{supp}(\omega)$ is sharp. (If some $f \in \mathrm{supp}(\omega)$ were not sharp then we could show, as in the proof of Theorem 4, that the case (1) holds.) We denote by C the support clone of W. We assume that none of the cases (2a), (2b), (2c), (3) (with $k = 3$) of the theorem applies as we would be done in any of these cases.

By Lemma 2, we can assume that ω assigns weight -1 to each of the three ternary projections and thus

$$\sum_{f \in \mathrm{supp}(\omega)} \omega(f) \;=\; 3\,. \tag{12}$$

Let $P_i \subseteq C$ be the Pixley operations of type $i \in \{1,2,3\}$ from C. Since C is a clone we have

$$|P_1| \;=\; |P_2| \;=\; |P_3|\,. \tag{13}$$

By Lemma 2, we have for any three related Pixley operations $p_1 \in P_1$, $p_2 \in P_2$, and $p_3 \in P_3$,

$$\omega(p_1) \;=\; \omega(p_2) \;=\; \omega(p_3)\,, \tag{14}$$

and

$$\sum_{p\in P_1}\omega(p) \;=\; \sum_{p\in P_2}\omega(p) \;=\; \sum_{p\in P_3}\omega(p)\,. \tag{15}$$

We set $P = P_1 \cup P_2 \cup P_3$ to be the set of all Pixley operations from C and $w(P) = \sum_{p\in P}\omega(p)$.

By Lemma 2, the same holds for the three types of ternary semiprojections. In particular, we denote by $S_i \subseteq C$ the operations from C that are semiprojections of type $i \in \{1,2,3\}$. Since C is a clone, we have

$$|S_1| \;=\; |S_2| \;=\; |S_3|\,. \tag{16}$$

For any three related semiprojections $s_1 \in S_1$, $s_2 \in S_2$, and $s_3 \in S_3$,

$$\omega(s_1) \;=\; \omega(s_2) \;=\; \omega(s_3)\,, \tag{17}$$

and

$$\sum_{s\in S_1}\omega(s) \;=\; \sum_{s\in S_2}\omega(s) \;=\; \sum_{s\in S_3}\omega(s)\,. \tag{18}$$

We set $S = S_1 \cup S_2 \cup S_3$ to be the set of all semiprojections from C and $w(S) = \sum_{s\in S}\omega(s)$.

To simplify the presentation, we use the same notation for weightings, index sets, etc. in the following three steps since the steps are similar (but independent). Thus, for example, when one reads J in Step II it refers to J defined in Step II and not in Step I.

Step I: Eliminating Pixley operations.

We now show how to eliminate Pixley operations if needed, that is, assume $w(P) > 0$ and thus some operations from P are assigned positive weight.

First assume that ω assigns positive weight to only Pixley operations, that is, $\mathrm{supp}(\omega) \subseteq P$. Hence $w(P) = 3$. Take arbitrary $p_1, p_1' \in P_1$, $p_2, p_2' \in P_2$, and $p_3, p_3' \in P_3$. The following claims can be verified from the definitions: $p_1[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, p_1']$ is a majority operation, $p_2[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, p_1'] = \mathrm{e}_1^{(3)}$, and $p_3[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, p_1'] = \mathrm{e}_2^{(3)}$. Consequently, $\omega[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, p_1']$ assigns weight -1 to p_1', $+1$ to majority operations, and 0 otherwise. Similarly, $\omega[\mathrm{e}_1^{(3)}, p_2', \mathrm{e}_3^{(3)}]$ assigns weight -1 to p_2', $+1$ to majority operations, and 0 otherwise. Finally, $\omega[p_3', \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}]$ assigns weight -1 to p_3', $+1$ to majority operations, and 0 otherwise. Overall, the weighting

$$\mu \;=\; \omega + \sum_{p\in P_1}\omega(p)\omega[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, p] + \sum_{p\in P_2}\omega(p)\omega[\mathrm{e}_1^{(3)}, p, \mathrm{e}_3^{(3)}] + \sum_{p\in P_3}\omega(p)\omega[p, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}] \tag{19}$$

assigns weight -1 to each of the three ternary projections and weight 3 to majority operations. By Lemma 3, the intermediate superpositions in (19) can be improper as long as the resulting weighting μ is indeed a weighting. Thus $\mu \in W$ and case (2a) of the theorem holds.

Let assume that $\mathrm{supp}(\omega) \not\subseteq P$, that is, ω assigns positive weight not only to Pixley operations. Let $J = \{(p_1, p_2, p_3) \in P_1 \times P_2 \times P_3 \mid p_1, p_2, p_3 \text{ are related}\}$ and $\bar{J} = \{(\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)})\} \cup J$.

We consider the following linear system: for all Pixley operations $p \in P$,

$$\sum_{(f,g,h)\in\bar{J}} x_{f,g,h}\,\omega[f,g,h](p) \;=\; 0\,. \tag{20}$$

By Gordan's Theorem, (20) has a nonzero nonnegative solution if and only if the following system of strict inequalities is unsatisfiable: for all $(f,g,h) \in \bar{J}$,

$$\sum_{p\in P} y_p\,\omega[f,g,h](p) \;>\; 0\,. \tag{21}$$

Consider the case $(f,g,h) = (\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)})$. Then $\omega[f,g,h] = \omega$ and thus $\omega[f,g,h](p) = \omega(p) > 0$ for all $p \in \mathrm{supp}(\omega) \cap P$, by the definition of P. Moreover, by (15)

$$\sum_{p \in P} y_p \omega(p) \;=\; \sum_{p_1 \in P_1} (y_{p_1} + y_{p_2} + y_{p_3}) \omega(p_1)\,, \tag{22}$$

where we denoted (with a slight abuse of notation) by p_2 and p_3 the related operations of p_1.

For $(f,g,h) = (\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)})$ the LHS of (21) is equal to (22). Thus, for (21) to hold in this case we must have at least one triple of related operations $(p_1, p_2, p_3) \in J$ with $y_{p_1} + y_{p_2} + y_{p_3} > 0$.

Suppose $(p_1, p_2, p_3) \in J$ is chosen to maximise $y_{p_1} + y_{p_2} + y_{p_3}$. The left-hand side of (21) when $(f,g,h) = (p_1, p_2, p_3)$ is equal to

$$- y_{p_1} - y_{p_2} - y_{p_3} + \sum_{s \in S} y_{s[p_1,p_2,p_3]} \omega(s)\,, \tag{23}$$

since $o[p_1, p_2, p_3]$ is equal to a Pixley operation only when o is one of the three projections, which give the first three terms in (23), or a semiprojection, which gives the last term of (23), the sum over S.

We have

$$\begin{aligned}
\sum_{s \in S} y_{s[p_1,p_2,p_3]} \omega(s) \;&=\; \sum_{s_1 \in S_1} y_{s_1[p_1,p_2,p_3]} \omega(s_1) + \sum_{s_2 \in S_2} y_{s_2[p_1,p_2,p_3]} \omega(s_2) + \sum_{s_3 \in S_3} y_{s_3[p_1,p_2,p_3]} \omega(s_3) \\
&=\; \sum_{\substack{(s_1,s_2,s_3) \in S_1 \times S_2 \times S_3 \\ s_1,s_2,s_3 \text{ related}}} (y_{s_1[p_1,p_2,p_3]} + y_{s_2[p_1,p_2,p_3]} + y_{s_3[p_1,p_2,p_3]}) \omega(s_1) \\
&\leq\; \frac{\omega(S)}{3} (y_{p_1} + y_{p_2} + y_{p_3}),
\end{aligned} \tag{24}$$

where the first equality follows from the definition of S; the second equality follows from fact that $s[p_1, p_2, p_3]$ is a Pixley operation of type i given s is a semiprojection of type i, where $i \in \{1, 2, 3\}$, and hence $(s_1[p_1, p_2, p_3], s_2[p_1, p_2, p_3], s_3[p_1, p_2, p_3])$ is a triple of related Pixley operations given that (s_1, s_2, s_3) is a triple of related semiprojections; and the last inequality follows from the definition of $w(S)$ and the choice of (p_1, p_2, p_3).

Combining (23) and (24), we have

$$- y_{p_1} - y_{p_2} - y_{p_3} + \sum_{s \in S} y_{s[p_1,p_2,p_3]} \omega(s) \;\leq\; -y_{p_1} - y_{p_2} - y_{p_3} + \frac{w(S)}{3} (y_{p_1} + y_{p_2} + y_{p_3}) \;<\; 0\,, \tag{25}$$

where the last strict inequality follows from $w(S) < 3$ since $w(P) > 0$ and (12).

Hence, (21) is unsatisfiable and, by Gordan's Theorem, (20) must have a nonzero nonnegative solution x^*. We finish Step I by using x^* to prove the existence of a weighting in W that assigns zero weight to all Pixley operations.

Let

$$\mu' = \sum_{(f,g,h) \in \bar{J}} x^*_{f,g,h} \omega[f,g,h]\,, \tag{26}$$

be a weighted sum of superpositions of ω.

By the choice of x^*, μ' assigns zero weight to all Pixley operations. From the definition of $\bar{J}$,

$$\mu' \;=\; x^*_{\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}} \omega[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}] + \sum_{(f,g,h) \in J} x^*_{f,g,h} \omega[f,g,h]\,. \tag{27}$$

We know that x^* is nonzero and nonnegative. Note that $x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}} > 0$ and $x^*_{f,g,h} = 0$ for all $(f,g,h) \in J$ would contradict that μ' assigns zero weight to all Pixley operations.

Let $(f,g,h) \in J$ and $i \in \{1,2,3\}$. Then $o[f,g,h]$ is a Pixley operation of type i if o is a semiprojection of type i, $o[f,g,h]$ is a semiprojection of type i if o is a Pixley operation of type i, $o[f,g,h]$ is a majority operation if o is a minority operation, and finally $o[f,g,h]$ is a minority operation if o is a majority operation. It follows that $x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}} = 0$ would contradict that μ' assigns zero weight all Pixley operations since Pixley operations of type i would be assigned negative weight since $w(S) - 3 < 0$ as $w(P) > 0$. Thus $x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}} > 0$ and $x^*_{f,g,h} > 0$ for at least one $(f,g,h) \in J$. Consequently, μ' is a nonzero weighting that assigns zero weight to all Pixley operations. By Lemma 3, $\mu' \in W$.

Step II: Eliminating semiprojections.

We now show how to eliminate semiprojections if needed. If μ' obtained in Step I assigns positive weight to only semiprojections then we are in case (3) (with k=3) of the theorem. Thus let assume that μ' assigns positive weight to semiprojections and at least one majority or minority operation.

By Lemma 2, we can assume the existence of a ternary weighting $\omega \in W$ which assigns weight -1 to all three projections (and thus assigns total positive weight 3). We will use the same notation for S, S_1, S_2, and S_3 as before although note that the weighting ω is now different from the one we had before and throughout Step I. We have $w(S) < 3$ and (17), (16), (18) still hold for ω.

We again use Gordan's Theorem to show that there exists a nonzero ternary weighting in W that assigns positive weight to majority and minority operations only.

Let $J = \{(s_1,s_2,s_3) \in S_1 \times S_2 \times S_3 \mid s_1,s_2,s_3 \text{ are related}\}$ and $\bar{J} = \{(\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)})\} \cup J$.

We consider the following linear system: for all semiprojections $s \in S$,

$$\sum_{(f,g,h)\in\bar{J}} x_{f,g,h}\,\omega[f,g,h](s) \;=\; 0\,. \tag{28}$$

By Gordan's Theorem, (28) has a nonzero nonnegative solution if and only if the following system of strict inequalities is unsatisfiable: for all $(f,g,h) \in \bar{J}$,

$$\sum_{s\in S} y_s\,\omega[f,g,h](s) \;>\; 0\,. \tag{29}$$

As in Step I, we can argue that (29) is unsatisfiable. Consider the case $(f,g,h) = (\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)})$. Then $\omega[f,g,h] = \omega$ and thus $\omega[f,g,h](s) = \omega(s) > 0$ for all $s \in \mathrm{supp}(\omega) \cap S$, by the definition of S. Moreover, by (18)

$$\sum_{s\in S} \omega(s) y_s \;=\; \sum_{s_1\in S_1} (y_{s_1} + y_{s_2} + y_{s_3})\omega(s_1)\,, \tag{30}$$

where we denoted by s_2 and s_3 the related operations of s_1.

Therefore, for (29) to hold when $(f,g,h) = (\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)})$ we must have at least one triple of related semiprojections $(s_1,s_2,s_3) \in J$ with $y_{s_1} + y_{s_2} + y_{s_3} > 0$.

Suppose $(s_1,s_2,s_3) \in J$ is chosen to maximise $y_{s_1} + y_{s_2} + y_{s_3}$. The left-hand side of (29) when $(f,g,h) = (s_1,s_2,s_3)$ is equal to

$$-\,y_{s_1} - y_{s_2} - y_{s_3} + \sum_{s\in S} y_{s[s_1,s_2,s_3]}\omega(s)\,, \tag{31}$$

since $o[s_1, s_2, s_3]$ is equal to a semiprojection only when o is one of the three projections, which give the first three terms in (31), or a semiprojection, which gives the last term of (31), the sum over S.

We have

$$
\begin{aligned}
\sum_{s\in S} y_{s[s_1,s_2,s_3]}\omega(s) &= \sum_{s_1\in S_1} y_{s_1[s_1,s_2,s_3]}\omega(s_1) + \sum_{s_2\in S_2} y_{s_2[s_1,s_2,s_3]}\omega(s_2) + \sum_{s_3\in S_3} y_{s_3[s_1,s_2,s_3]}\omega(s_3) \\
&= \sum_{\substack{(s_1,s_2,s_3)\in S_1\times S_2\times S_3 \\ s_1,s_2,s_3 \text{related}}} (y_{s_1[s_1,s_2,s_3]} + y_{s_2[s_1,s_2,s_3]} + y_{s_3[s_1,s_2,s_3]})\omega(s_1) \\
&\leq \frac{\omega(S)}{3}(y_{s_1} + y_{s_2} + y_{s_3}),
\end{aligned}
\tag{32}
$$

where the first equality follows from the definition of S; the second equality follows from fact that $s[s_1, s_2, s_3]$ is a semiprojection of type i given s is a semiprojection of type i, where $i \in \{1, 2, 3\}$, and hence $(s_1[s_1, s_2, s_3], s_2[s_1, s_2, s_3], s_3[s_1, s_2, s_3])$ is a triple of related semiprojections given that (s_1, s_2, s_3) is a triple of related semiprojections; and the last inequality follows from the definition of $w(S)$ and the choice of (s_1, s_2, s_3).

Combining (31) and (32), we have

$$
- y_{s_1} - y_{s_2} - y_{s_3} + \sum_{s\in S} y_{s[s_1,s_2,s_3]}\omega(s) \;\leq\; -y_{s_1} - y_{s_2} - y_{s_3} + \frac{w(S)}{3}(y_{s_1} + y_{s_2} + y_{s_3}) \;<\; 0\,, \tag{33}
$$

where the last strict inequality follows from $w(S) < 3$.

Hence, (29) is unsatisfiable and, by Gordan's Theorem, (28) must have a nonzero nonnegative solution x^*. We finish Step II by using x^* to prove the existence of a weighting in W that assigns zero weight to all semiprojections.

Let

$$
\mu = \sum_{(f,g,h)\in J} \omega[f, g, h]x^*_{f,g,h}\,, \tag{34}
$$

be a weighted sum of superpositions of ω. By the choice of x^*, μ assigns zero weight to all semiprojections.

From the definition of $\bar{J}$,

$$
\mu \;=\; x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}}\omega[\mathrm{e}_1^{(3)}, \mathrm{e}_2^{(3)}, \mathrm{e}_3^{(3)}] + \sum_{(f,g,h)\in J} x^*_{f,g,h}\omega[f, g, h]\,. \tag{35}
$$

We know that x^* is nonzero and nonnegative. Note that $x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}} > 0$ and $x^*_{f,g,h} = 0$ for all $(f, g, h) \in J$ would contradict that μ' assigns zero weight to all semiprojections.

Let $(s_1, s_2, s_3) \in J$ and $i \in \{1, 2, 3\}$. Then $o[s_1, s_2, s_3]$ is a semiprojection of type i if o is a semiprojection of type i, $o[s_1, s_2, s_3]$ is a majority operation if o is a majority operation, and finally $o[s_1, s_2, s_3]$ is a minority operation if o is a minority operation. (Note that we do not need to consider the case when f is a Pixley operation as ω assigns zero weight to all Pixley operations.) It follows that $x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}} = 0$ would contradict that μ assigns zero weight all semiprojections since semiprojections of type i would be assigned negative weight since $w(S)-3 < 0$ as ω assigns positive weight to some majority or minority (or possibly both) operations. Thus $x^*_{\mathrm{e}_1^{(3)},\mathrm{e}_2^{(3)},\mathrm{e}_3^{(3)}} > 0$ and $x^*_{f,g,h}$ for at least one $(f, g, h) \in J$. Consequently, μ is a nonzero weighting that assigns zero weight to all semiprojections. By Lemma 3, $\mu \in W$.

Step III: Majority and minority operations.

In Steps I and II, we have shown that any weighted clone W with a positive ternary weighting contains a ternary weighting that assigns nonzero weight to semiprojections alone (case (3) of the theorem with $k = 3$), or a mix of majority and minority operations. Finally, we will show that if W contains a weighting ω that assigns weight to majority and minority operations alone then W also contains a weighting of one of the three types described in cases (2a), (2b), and (2c) of the theorem.

Let M_1 and M_2 denote the sets of majority and minority operations in the support of ω. Suppose that ω assigns total weight $2 + a$ to M_1 and total weight $1 - a$ to M_2 for some $a > 0$. For each $f \in M_2$, we define $\mu_f = \omega[\mathrm{e}_1^{(3)}, \mathrm{e}_1^{(3)}, f]$, so μ_f assigns weight a to $\mathrm{e}_1^{(3)}$ and weight $-a$ to f. Note that μ_f is not a proper weighting, since $a > 0$. We obtain a weighting $\mu \in W$ which assigns positive weight to majority operations only as follows:

$$\mu = \omega + \sum_{f \in M_2} \frac{\omega(f)}{a} \mu_f \,. \tag{36}$$

Similarly, suppose that ω assigns total weight $2-a$ to M_1 and total weight $1+a$ to M_2 for some $a > 0$. For each $f \in M_1$, we define $\mu_f = \omega[\mathrm{e}_1^{(3)}, f, f]$, so μ_f assigns weight a to $\mathrm{e}_1^{(3)}$ and weight $-a$ to f. We obtain a weighting $\mu \in W$ which assigns positive weight to minority operations only as follows:

$$\mu = \omega + \sum_{f \in M_1} \frac{\omega(f)}{a} \mu_f \,. \tag{37}$$

In both cases $\mu \in W$ by Lemma 3. □

5 Conclusions

We have presented new results on the structure of weighted clones that delimit the possibilities for tractable valued constraint languages. In order to establish our results, we have presented a novel technique for ruling out certain types of operations from the support of a given weighting. The method considers certain extreme cases of the dual of the linear program that demonstrates the existence of a weighted sum of superpositions that assigns zero weight to the forbidden operations. We believe that our results and techniques will prove useful in further studies of the structure of weighted clones. However, understanding the structure of weighted clones appears a difficult problem in general. For instance, whilst the computational complexity of finite-valued constraint languages is well understood [46], the structure of the corresponding weighted clones is not, as discussed in Section 3.

In recent work on the tractability of valued constraint languages, it has been shown that a necessary condition for tractability is the existence of a *cyclic* weighted polymorphism [35].[7] Moreover, it has been also shown that, under the assumption of the dichotomy conjecture of Feder and Vardi for the decision problem, this condition is also sufficient [32].

Acknowledgements

The authors are grateful to Páidí Creed and Peter Fulla for valuable discussions.

[7]A k-ary operation if *cyclic* if $f(x_1, x_2, \ldots, x_k) = f(x_2, \ldots, x_k, x_1)$ for every $x_1, \ldots, x_k$. A weighting ω is cyclic if every operation $f \in \mathrm{supp}(\omega)$ is cyclic.

References

[1] Krzysztof Apt. *Principles of Constraint Programming.* Cambridge University Press, 2003.

[2] Libor Barto and Marcin Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*, 61(1), 2014. Article No. 3.

[3] Libor Barto, Marcin Kozik, and Todd Niven. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM Journal on Computing*, 38(5):1782–1802, 2009.

[4] Manuel Bodirsky, Hubie Chen, Jan Kára, and Timo von Oertzen. Maximal infinite-valued constraint languages. *Theoretical Computer Science*, 410(18):1684–1693, 2009.

[5] Andrei Bulatov. A Graph of a Relational Structure and Constraint Satisfaction Problems. In *Proceedings 19th IEEE Symposium on Logic in Computer Science (LICS'04)*, pages 448–457. IEEE Computer Society, 2004.

[6] Andrei Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.

[7] Andrei Bulatov, Andrei Krokhin, and Peter Jeavons. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

[8] Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic*, 12(4), 2011. Article 24.

[9] Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons. The complexity of maximal constraint languages. In *Proceedings 33rd ACM Symposium on Theory of Computing (STOC'01)*, pages 667–674, 2001.

[10] David A. Cohen, Martin C. Cooper, Páidí Creed, Peter Jeavons, and Stanislav Živný. An algebraic theory of complexity for discrete optimisation. *SIAM Journal on Computing*, 42(5):915–1939, 2013.

[11] David A. Cohen, Martin C. Cooper, Peter Jeavons, and Stanislav Živný. Dualisation via binarisation for valued constraints. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2015.

[12] David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.

[13] Páidí Creed and Stanislav Živný. On minimal weighted clones. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11)*, volume 6876 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 2011.

[14] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.

[15] Béla Csákány. Minimal clones – a minicourse. *Algebra Universalis*, 54(1):73–89, 2005.

[16] Rina Dechter. *Constraint Processing.* Morgan Kaufmann, 2003.

[17] Klaus Denecke and Shelly L. Wismath. *Universal Algebra and Applications in Theoretical Computer Science*. Chapman and Hall/CRC Press, 2002.

[18] Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[19] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[20] Pavol Hell and Jaroslav Nešetřil. On the Complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

[21] Pavol Hell and Jaroslav Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.

[22] D. Hobby and R.N. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1988.

[23] Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing*, 43(3):1064–1084, 2014.

[24] Peter Jeavons, Andrei Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 113:21–55, 2014.

[25] Peter G. Jeavons. On the Algebraic Structure of Combinatorial Problems. *Theoretical Computer Science*, 200(1-2):185–204, 1998.

[26] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. Closure Properties of Constraints. *Journal of the ACM*, 44(4):527–548, 1997.

[27] Peter Jonsson, Fredrik Kuivinen, and Gustav Nordh. MAX ONES Generalized to Larger Domains. *SIAM Journal on Computing*, 38(1):329–365, 2008.

[28] Peter Jonsson and Gustav Nordh. Introduction to the MAXIMUM SOLUTION Problem. In *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 255–282. Springer, 2008.

[29] Peter Jonsson, Gustav Nordh, and Johan Thapper. The maximum solution problem on graphs. In *Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science (MFCS'07)*, volume 4708 of *Lecture Notes in Computer Science*, pages 228–239. Springer, 2007.

[30] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001.

[31] Vladimir Kolmogorov. The power of linear programming for finite-valued CSPs: a constructive characterization. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 625–636. Springer, 2013.

[32] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The complexity of general-valued CSPs. Technical report, 2015. arXiv:1502.07327.

[33] Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. The power of linear programming for general-valued CSPs. *SIAM Journal on Computing*, 44(1):1–36, 2015.

[34] Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. *Journal of the ACM*, 60(2), 2013. Article No. 10.

[35] Marczin Kozik and Joanna Ochremiak. Algebraic properties of valued constraint satisfaction problem. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, Lecture Notes in Computer Science. Springer, 2015.

[36] R.N. McKenzie, G.F. McNulty, and W.F. Taylor. *Algebras, Lattices and Varieties*, volume I. Wadsworth and Brooks, California, 1987.

[37] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[38] Justin K. Pearson and Peter G. Jeavons. A survey of tractable constraint satisfaction problems. Technical Report CSD-TR-97-15, Royal Holloway, University of London, July 1997.

[39] Robert Powell and Andrei A. Krokhin. A reduction from valued CSP to min cost homomorphism problem for digraphs. Technical report, 2015. arXiv:1507.01776.

[40] Thomas J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226. ACM, 1978.

[41] Stanisław Świerczkowski. Algebras which are independently generated by every n elements. *Fundamenta Mathematicae*, 49:93–104, 1960.

[42] A. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminaires de Mathematiques Superieures*. University of Montreal, 1986.

[43] Rustem Takhanov. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*, pages 657–668, 2010.

[44] Rustem Takhanov. Extensions of the Minimum Cost Homomorphism Problem. In *Proceedings of the 16th International Computing and Combinatorics Conference (COCOON'10)*, volume 6196 of *Lecture Notes in Computer Science*, pages 328–337. Springer, 2010.

[45] Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*, pages 669–678. IEEE, 2012.

[46] Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13)*, pages 695–704. ACM, 2013. Full version available at arXiv:1210.2977v3.

[47] Johan Thapper and Stanislav Živný. Sherali-Adams relaxations for valued CSPs. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 1058–1069. Springer, 2015.

[48] Hannes Uppman. The Complexity of Three-Element Min-Sol and Conservative Min-Cost-Hom. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP'13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 804–815. Springer, 2013.

[49] Hannes Uppman. Computational Complexity of the Extended Minimum Cost Homomorphism Problem on Three-Element Domains. In *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS'14)*, volume 25, pages 651–662, 2014.

[50] Stanislav Živný. *The complexity of valued constraint satisfaction problems.* Cognitive Technologies. Springer, 2012.